Computationally Efficient Graph Matching via Energy Vector Extraction

Abstract. This paper presents a method for graph matching based on domain knowledge by quantifying representative graph features. Our method searches and extracts the most relevant cues in different graphs. Once these cues are extracted and quantified, a new energy function is used to match the different graphs based on the obtained features values. This approach has been successfully applied for deformable template matching. As a result the error of matching is reduced, as well as the computational cost by efficiently selecting and grouping representative features.

1 Introduction

Graphs hold a great representative power which makes them the most natural way to encode and symbolize any pattern [1]. All non-verbal human communications, ranging from hand gestures to written documents, include some concept of a graph. Many applications in computer vision include selecting and extracting information from graphs, since graphs provide a useful way to represent knowledge. For this reason Tombre proposed a method for analyizing engineering drawings [2], and Kruger, Potzsch, and Andmalsburg proposed a method for determining face position and pose based on labeled graphs [3].

Due to this power, graph matching is a logical way to compare and recognize different objects. The main challenge in any graph matching method is to extract representative segments of different graphs for comparison. Many approaches have been considered: some were based on one-to-one correspondence [1], while others were based on many-to-many correspondence [4]. However, these methods usually suffered from computational complexity and inefficiency while handling distortions.
In this paper, we present a method for graph matching based on feature extraction to obtain a vector representation for each graph which is applied for deformable template recognition. A new energy function is formulated to extract this vector based on the graph attributes. Then, a simple distance function is applied for the final

matching. This method is robust to noise, segmentation errors, graphs with missing or extra edges, and graphs with missing or extra node.

This paper is organized as follows. Section 2 discusses briefly related works. Section 3 presents the feature extraction method while section 4 presents the matching method. In section 5, experimental results are discussed. Concluding remarks are available in section 6.

2 Related Work

Graph matching is usually addressed as a method for object recognition. Gold and Rangarajan [1] propose a one-to-one correspondence is established through a graduated assignment Kim and Kak [5] propose a bipartite graph matching using discrete relaxation for 3-D object reconstruction. Other algorithm uses a quadratic programming approach for graph matching using a maximal clique framework [6]. Many-to-many correspondence has been also investigated [4], frequently in the context of edit-distance, [7].

However, these methods are either computationally complex or does not support slight distortions in graphs. So, a small deformation in the graph to be matched introduces a lot of error in the matching result, unless the algorithm is computationally expensive.

In this paper, a robust and low computational cost method is introduced. Its strength lies in its ability to assign quantitative values to representative graph features such that the matching process is only a simple distance function.

3 Energy Function

A graph G = (N, E) can be interpreted as a set of basic forms which are nodes and edges. Figure 1 shows an interpretation of a given object as a graph. The features used in matching different graphs are the information obtained from the nodes and edges.

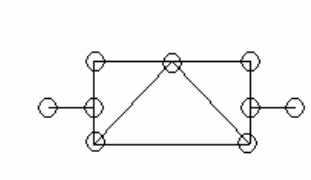

Fig. 1. Object representation as a set of nodes and edges

This method transforms the graph into a vector of energy values. An energy function is calculated for each node in the graph. This energy function aims at encoding the number of edges connected to each node, their relative length, their shape, and the average relative angle of each node. Then, each graph is represented as a vector of energy values calculated at each of its nodes.

Towards this end, we define a novel energy function as follows:

$$\mathrm{E}(i) = \sum_{p=1}^{n} p * k1 + \frac{\sum_{p=1}^{n} L_{ip}}{\sum_{j=1}^{m}\sum_{p=1}^{n} L_{jp}} * k2 + \frac{\sum_{p=1}^{n} \theta_{ip}}{n} * \frac{k3}{2.\pi} + \frac{\sum_{p=1}^{n} A_{ip}}{\sum_{j=1}^{m}\sum_{p=1}^{n} A_{jp}} * k4 \quad (1)$$

where n is the number of edges attached to node i, m is the number of nodes in the graph, L_{ip} is the length of the pth edge attached to node i, θ_{ip} is the angle the pth edge form with the horizontal, A_{ip} is the area enclosed by the edge in case it is not a straight line. This area is representative of the shape of the edge as shown in Fig. 2.

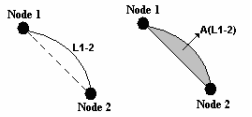

Fig. 2. Area enclosed by a circular edge between Node 1 and Node 2

Lastly, k1, k2, k3, and k4 are weights based on the characteristics of the considered graphs.

4 Graph Matching

Let's assume a graph $\mathbf{G_k}$ is to be matched with a set of graphs $\mathbf{G_1}$,$\mathbf{G_2}$,…,$\mathbf{G_n}$. Once the energy vectors $\mathbf{E_k}$ and $\mathbf{E_1}$,$\mathbf{E_2}$,…,$\mathbf{E_n}$ are obtained, they are then sorted in descending order. Then, the distance between $\mathbf{E_k}$ and each of the other energy vectors is calculated according to Eq. 2.

$$d_j = \sum_{i=1}^{n} | E_k(i) - E_j(i) | \quad (2)$$

where d_j is the distance between G_k and G_j, n is the maximum number of nodes found in G_k and G_j .

4.1 Robustness

This method is robust to perturbations in graph structure due to the fact that the energy function, Eq. 1, emphasizes the similarities between graphs. The idea is to compare the most representative features of one graph to the most representative features of the second graph. Therefore, sorting the two energy vector, and comparing them one-by-one (the first value in $\mathbf{E_1}$ to the first value in $\mathbf{E_2}$, the second value in $\mathbf{E_1}$ to the second value of $\mathbf{E_2}$, and so on) provide a very simple and effective way to compare the representative features. Any noise or distortion introduced will have low energy value and will appear at the end of the energy vector; therefore, it will have a minimal effect on the matching process. Also, this matching process is independent on the size of the graphs, since the relative lengths of the edges are considered. Matching is also independent on the position of the graphs since relative angles are considered.

4.2 Efficiency

The main advantage of this method is its computational efficiency. It is able to achieve robust detection in O(n) where *n* is the maximum number of nodes found in the different graphs. This is considered a great achievement when compared with other methods that are considered low-cost yet they run in O(lm), where *l* and *m* are the number of links in the two graphs [1].

5 Experimental Results

We worked with a database[1] of four graphs and 500 distorted versions of these graphs. The four graphs are shown in Fig. 3 and their corresponding distorted versions are shown in Fig. 4.

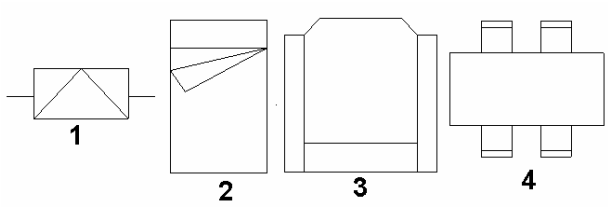

Fig. 3. The four basic graphs

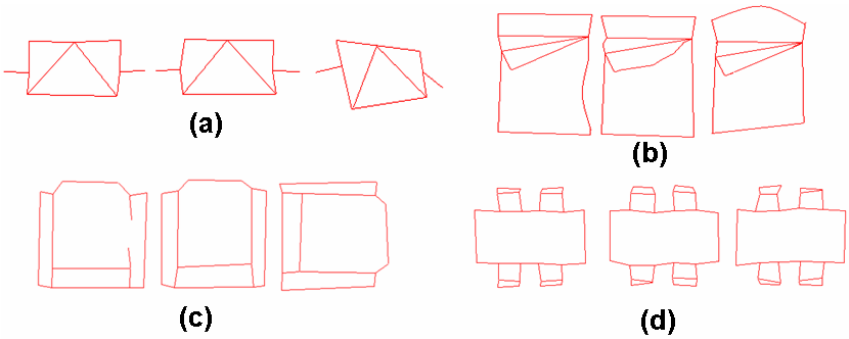

Fig. 4. (a) Distorted versions of Graph 1 (b) Distorted versions of Graph 2 (c) Distorted versions of Graph 3 (d) Distorted versions of Graph 4

[1] International Symbol Recognition Contest at GREC 2003, Barcelona, Catalonia, Spain, July 29, 2003

The purpose was to match each distorted graph with its original version. Setting experimentally k1 to 10, k2 to 5, k3 to 3, and k4 to 3, we were able to achieve a 100% recognition rate. Table 1 shows the result of calculating the distance function between 16 distorted versions of Graph 1 and all four graphs. It can be easily seen that the minimum distance is produced when comparing each one (on each row) with Graph 1 indicating that the matching was done correctly. Tables 2, 3, and 4 show the same results for Graphs 2, 3, and 4 respectively.

TABLE 1
Distance Calculation for 16 Distorted Versions of Graph 1

(1-(a))	(2-(a))	(3-(a))	(4-(a))
9	**106**	**157**	**663**
12	**107**	**162**	**670**
12	**103**	**158**	**666**
8	**101**	**154**	**662**
10	**103**	**154**	**671**
11	**106**	**154**	**668**
11	**107**	**162**	**670**
9	**108**	**159**	**668**
10	**110**	**160**	**669**
12	**110**	**160**	**664**
8	**112**	**158**	**665**
8	**104**	**154**	**670**
12	**106**	**169**	**669**
11	**111**	**154**	**663**
10	**106**	**153**	**667**
9	**104**	**161**	**670**

TABLE 2
Distance Calculation for 16 Distorted Versions of Graph 2

(1-(b))	(2-(b))	(3-(b))	(4-(b))
94	**8**	**172**	**660**
96	**12**	**176**	**664**
94	**8**	**172**	**660**
96	**12**	**176**	**664**
96	**12**	**176**	**663**
94	**8**	**172**	**660**
96	**9**	**172**	**660**
94	**8**	**176**	**663**
97	**11**	**174**	**665**
97	**10**	**174**	**663**
95	**11**	**175**	**660**
94	**8**	**176**	**662**
95	**9**	**175**	**664**
96	**10**	**173**	**660**
94	**8**	**175**	**664**
94	**8**	**174**	**663**

TABLE 3
Distance Calculation for 16 Distorted Versions of Graph 3

(1-(c))	(2-(c))	(3-(c))	(4-(c))
158	177	8	572
155	178	4	567
155	176	5	571
157	180	3	569
155	176	5	571
157	180	8	569
158	180	7	567
155	176	3	572
156	178	4	571
154	177	6	569
155	176	7	571
158	180	8	570
156	177	7	571
156	180	3	568
158	177	3	572
157	176	5	576

TABLE 4
Distance Calculation for 16 Distorted Versions of Graph 4

(1-(d))	(2-(d))	(3-(d))	(4-(d))
667	676	571	33
667	686	577	31
669	678	569	23
670	679	570	24
670	679	570	28
668	675	568	24
670	678	571	17
669	678	570	22
668	679	569	25
671	675	568	29
669	676	573	23
670	684	567	19
667	682	568	28
670	680	572	30
669	679	570	26
671	677	569	25

6 Conclusions

In this paper, we present a computationally efficient graph matching technique based on feature extraction via energy vectors. This method has a very low order computational complexity O(n) and it is extremely robust to noise and distortion. The experimental results were impressive. In future, this method will be tested with more graphs specifically those including non-linear edges.

References

[1] Gold, S. and Rangarajan, A., "Graph Matching by Graduated Assignment*", IEEE Computer Society Conference on Computer Vision and Pattern Recognition(VCPR'96)*, pp. 239-244, June 1996. San Francisco, CA, USA.

[2] Tombre, K., "Analysis of Engineering Drawings: State of the Art and Challenges*", Graphics Recognition Algorithms and Systems, LNCS*, vol. 1389, pp. 257-264, 1998.

[3] Kruger, N., Potzsch, M., Andmalsburg, C.V.D., "Determination of Face Position and Pose with a Learned Representation Based on Labeled Graphs*", Image Vision Computing*, n. 15, pp. 665-673, 1997.

[4] Keselman, Y., Shokoufandeh, A., Demirci, M.F. and Dickinson, S., "Many-to-Many Graph Matching via Metric Embedding", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1-850 – 1-857, June 2003.

[5] Kim, W., and Kak, A.C., "3D object recognition using bipartite matching embedded in discrete relaxation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):224-251, 1991.

[6] Pelillo, M., Siddiqi, K., and Zucker, S., "Matching hierarchical structures using association graphs", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105-1120, November 1999.

[7] Liu, T.L., and Geiger, D., "Approximate tree matching and shape similarity", *7th International Conference on Computer Vision*, pp. 456-462, 1999.